Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1181129, 12 pages
https://doi.org/10.1155/2021/1181129

Research Article

Association Analysis of Private Information in Distributed Social Networks Based on Big Data

Dongning Jia,[1,2] **Bo Yin**,[1,2] **and Xianqing Huang**[2]

[1]*Ocean University of China, Qingdao, Shandong 266100, China*
[2]*Pilot National Laboratory for Marine Science and Technology (Qingdao), Qingdao, Shandong 266237, China*

Correspondence should be addressed to Bo Yin; ybfirst@ouc.edu.cn

Received 29 April 2021; Revised 19 May 2021; Accepted 21 May 2021; Published 7 June 2021

Academic Editor: Yuanpeng Zhang

As people's awareness of the issue of privacy leakage continues to increase, and the demand for privacy protection continues to increase, there is an urgent need for some effective methods or means to achieve the purpose of protecting privacy. So far, there have been many achievements in the research of location-based privacy services, and it can effectively protect the location privacy of users. However, there are few research results that require privacy protection, and the privacy protection system needs to be improved. Aiming at the shortcomings of traditional differential privacy protection, this paper designs a differential privacy protection mechanism based on interactive social networks. Under this mechanism, we have proved that it meets the protection conditions of differential privacy and prevents the leakage of private information with the greatest possibility. In this paper, we establish a network evolution game model, in which users only play games with connected users. Then, based on the game model, a dynamic equation is derived to express the trend of the proportion of users adopting privacy protection settings in the network over time, and the impact of the benefit-cost ratio on the evolutionarily stable state is analyzed. A real data set is used to verify the feasibility of the model. Experimental results show that the model can effectively describe the dynamic evolution of social network users' privacy protection behaviors. This model can help social platforms design effective security services and incentive mechanisms, encourage users to adopt privacy protection settings, and promote the deployment of privacy protection mechanisms in the network.

1. Introduction

Online social networks are changing people's daily behaviors, bringing great convenience to people's lives. With the frequent occurrence of privacy leaks, users have paid more and more attention to protecting the security of personal social data; the advent of the era of big data makes users' privacy and security face more threats. Therefore, the privacy and security issues of social networks have become a hot spot for users, service providers, and researchers [1, 2]. Relevant research work has proposed many methods and technologies to solve the problem of privacy leakage [3, 4]. Although it can alleviate this problem to a certain extent, it cannot completely eliminate the problem of privacy leakage. The main reason is that these studies have neglected the social network service provider's response to users. The root of this threat is the centralized social network service structure. In the centralized service model, online social network service providers are the core of the entire system architecture, and users are deprived of the right to control personal data. All user data is exposed to social network service providers, which provides necessary conditions for service providers to collect user data [5].

Privacy protection is different from traditional access control technology and encryption technology in that it does not arbitrarily cut off the access channel of secret data, nor does it simply decode the data. With the rapid changes in network technology, the scope of sensitive data content is also changing, from the table structure in the earliest relational model to the later streaming data and social network data containing sensitive data. Social network has a huge user group and data volume. Therefore, social network has become a hot research object in many disciplines such as computer science, sociology, and psychology, and social

network analysis has also become an important branch of Web data mining [6]. The development of social networks has made relevant data sets easier to obtain, and the development of social network analysis has also increased the possibility of data privacy information leakage. Therefore, the availability of data and the privacy of information are the trade-offs for data release. Therefore, it is very necessary to study the privacy protection of social networks.

Based on the observation and analysis of the traditional differential privacy protection mechanism, this paper verifies the effectiveness of differential privacy and analyzes the necessity and infeasibility of differential privacy on social networks. The subject designed a differential privacy mechanism based on interactive social networks, clarified in detail the advantages of improved interactive differential privacy compared to traditional differential privacy and its feasibility on social networks, and conducted a strong verification of its effectiveness. It focuses on the algorithm flow of the interactive differential privacy mechanism. This paper establishes a network evolution game model based on the social network structure and models the evolution process of users' privacy setting behaviors in online social networks. The experimental results show that the cost-benefit ratio of adopting a privacy mechanism has an important impact on the deployment and implementation of the privacy mechanism in social networks. The experimental results also show that the model proposed in this paper can effectively portray user privacy protection behaviors in social networks.

2. Related Work

Node attribute values are divided into identification information, quasi-identification information, and sensitive information [7]. Identification information is an attribute that can explicitly indicate an individual's identity, such as name and ID number; quasi-identification information can implicitly indicate an individual's identity, such as age and gender. Generally, a combination of multiple quasi-identification information is required to indicate an individual's identity. Sensitive information is information that needs to be protected, such as personal income and personal medical conditions. If the attacker knows the node attribute value he owns and matches the public social network data, it is possible to identify the true identity of the node and then obtain the user's sensitive information. At present, the attacker mainly matches and recognizes the nodes in the social network based on the background knowledge of some attack targets, so as to accurately or with a certain probability to identify the location of the attack target in the social network [8]. The process of the attacker matching and identifying the location of the attack target based on background knowledge is called node reidentification. For example, the nodes in the social network can be filtered according to the attribute values of multiple nodes, thereby further reducing the attack range of the attacker and increasing the attack hit rate [9].

Relevant scholars discussed how to implement the node k-anonymity model in social networks where each node has attribute information [10]. Gender and other generalization operations are performed, and k nodes with the same quasi-identity attributes are divided into the same cluster after generalization, so that the attacker's hit rate of attacks based on quasi-identification information is reduced to $1/k$. Related scholars apply the (k, l) model of the database to the field of social network privacy protection [11]. It requires that on the basis of the k-anonymity model, there must be at least l users with different information in the cluster to ensure that the network resists the k-anonymity model attack. Related scholars proposed a k-degree anonymous model for node degree, which requires the number of nodes with the same degree attribute in the network to be greater than or equal to k [12]. At the same time, the k-degree anonymity requirement is achieved by adding extra edges. Researchers propose a k-neighborhood model to resist neighborhood attacks [13]. The model requires that each node has at least $k-1$ nodes with the same neighborhood structure and uses methods such as adding pseudo edges in the specific implementation.

Data perturbation mainly randomizes and modifies the original social network graph, so that the attacker cannot identify the target node based on the background knowledge he has mastered. At present, the general methods for realizing data disturbance include randomly adding pseudo edges or pseudo nodes, deleting nodes or edges, and modifying the attributes of nodes or edges. The edge weight represents the strength of the relationship between two nodes. Relevant scholars proposed the edge weight protection technology using greedy strategy in weighted undirected graphs to modify the weights of key edges under the premise of ensuring that the shortest path does not change, maintain the overall structure of the network before and after anonymity, and minimize the amount of information loss [14]. However, this article only considers the edge weight information and cannot resist other link attacks. Related scholars have evaluated Twitter's privacy policy and proposed the Hummingbird structure [15]. Hummingbird has made some changes on Twitter, which can protect the content of tweets and prevent hashtags from being obtained by centralized servers. However, these studies have always been based on the complete trust of centralized servers. In fact, centralized servers also have certain security risks and may leak user privacy. Although some studies encrypt and protect data on the server side, it still cannot prevent OSN service providers from monitoring user interactions, censoring or deleting user data, or even controlling who can establish social relationships with a social circle. Distributed structure does not rely on centralized servers, but it also faces a series of challenges. For example, encryption algorithms can ensure confidentiality and integrity while ensuring that they can provide high efficiency like centralized server structures.

Related scholars have proposed a multilevel security method [16]. In this method, trust is just a parameter used to determine the security level of a visitor or resource owner. Semidistributed autonomous access control was later proposed. It is an execution mechanism used to control information sharing in online social networks. This mechanism can also standardize the access rules of network resources. This is mainly achieved through different levels of authority between users. These levels are based on the type of social

relationship, the depth of the social relationship, and the security level. Researchers have proposed an online social network access control mechanism based on Web semantic technology, which is scalable and fine-grained access control [17]. Its main idea is to encode information related to social networks through ontology encoding technology. These studies are based on traditional access control methods, but these researchers have overlooked a problem: traditional access control may also leak the privacy of users' social attributes. Later, a user social attribute privacy protection scheme based on node splitting was proposed [18]. It increased the anonymity of the original node by assigning the attribute links and social links of the original node to the new node, thereby protecting the user's sensitive attributes from disclosure. At the same time, it also splits the sensitive social attributes of users according to the different degrees of influence of the social network structure on the distribution of social attributes and the correlation between social attributes [19, 20].

3. Social Network Big Data Analysis Platform

3.1. Overall Design of Social Network Big Data Analysis. For the same social network data, algorithms in different fields can be used for research from many angles. Increasing the utilization rate of data can also speed up upper-level research work through the unified interface of the platform and reduce the code strength in data acquisition and processing. From a functional point of view, the design and implementation of such a platform need to meet the following requirements:

(1) Because users in social networks generate a large amount of data in real time, these data on the one hand supplement the user's historical data, and on the other hand, it completes the data of the entire relationship network. Therefore, incremental data acquisition capabilities for social networks are indispensable for the platform

(2) When mass data is incorporated into a platform for unified management, these data usually need to have a fixed format or mode to facilitate the processing of upper-level applications. At the same time, in the context of big data, it is necessary to prevent the loss of data as much as possible. When the loss inevitably occurs, the platform also needs to be able to provide enough copies for data recovery, so an independent module is needed in the platform

(3) It can provide fast calculation and algorithm expansion capabilities for massive data. In the massive data scenario, the ability to process data is a key factor in determining the availability of the platform, and whether the existing algorithms in the platform can be simply and efficiently extended is an important criterion for measuring the scalability of the platform. Therefore, the platform, as the manager of the entire cluster computing resources, needs to provide fast computing services and algorithm extension interfaces for massive data

(4) In the platform, not only need to understand the operating status of each component of the platform but also need to be able to easily view the progress and results of data analysis and algorithm operation. Therefore, visually displaying the status of all platforms through a unified interface not only facilitates the management and control of the platform but also increases the encapsulation of system modules so that users do not need to know the details of the platform

The social network big data analysis platform is built with Spark as the core and includes a one-stop platform for data acquisition, data processing, data mining, and data visualization. It has good openness, scalability, and versatility. The architecture of the system is shown in Figure 1, which mainly includes four modules: data capture, data preprocessing and storage, data mining and analysis, and data visualization.

3.2. Distributed Social Network Crawler Workflow. The reason why the crawler system adopts a distributed design is to improve the crawling efficiency of the crawler, but it also increases the difficulty of the system's task allocation, message communication, and error recovery. The distributed crawler system is functionally divided into a Master node and a Slave node. The main task of the Master node is to manage the status of the entire crawler system and schedule tasks, and the Slave node is responsible for crawling and parsing the assigned web links.

When the crawler successfully completes a crawling link, it will parse out the Weibo data or relational data saved in the current link. At this time, the crawler will send these data in the form of a message to the data collection component that exists on the Master side. If all the crawl links of a user are completed, a crawl task is completed. At this time, the data collection component will persist all the user data to disk and extract a user ID to be crawled from the user crawl queue.

When a Slave completes the registration process, the Master will extract a user ID to be crawled from the user crawling queue and create a crawling task. When this task is executed, the number of Weibo pages and the number of following pages of the user currently to be crawled will be spliced into URLs and placed in the crawl link queue. If the crawler sends a crawling task request, it will judge whether to send back the crawling task or add the crawler to the waiting queue according to whether the current crawling queue is empty. If the crawler sends the crawling exception information, the crawling link assigned to the crawler will be put into the crawling link queue again while waiting for the crawler to recover from the exception. If the crawler sends the crawling information, it will judge whether to send back the crawling task or add the crawler to the waiting queue according to whether the current crawling queue is empty. The crawler crawling process is shown in Figure 2.

3.3. Data Preprocessing and Analysis Module. HDFS is an open-source distributed file system implemented in Hadoop.

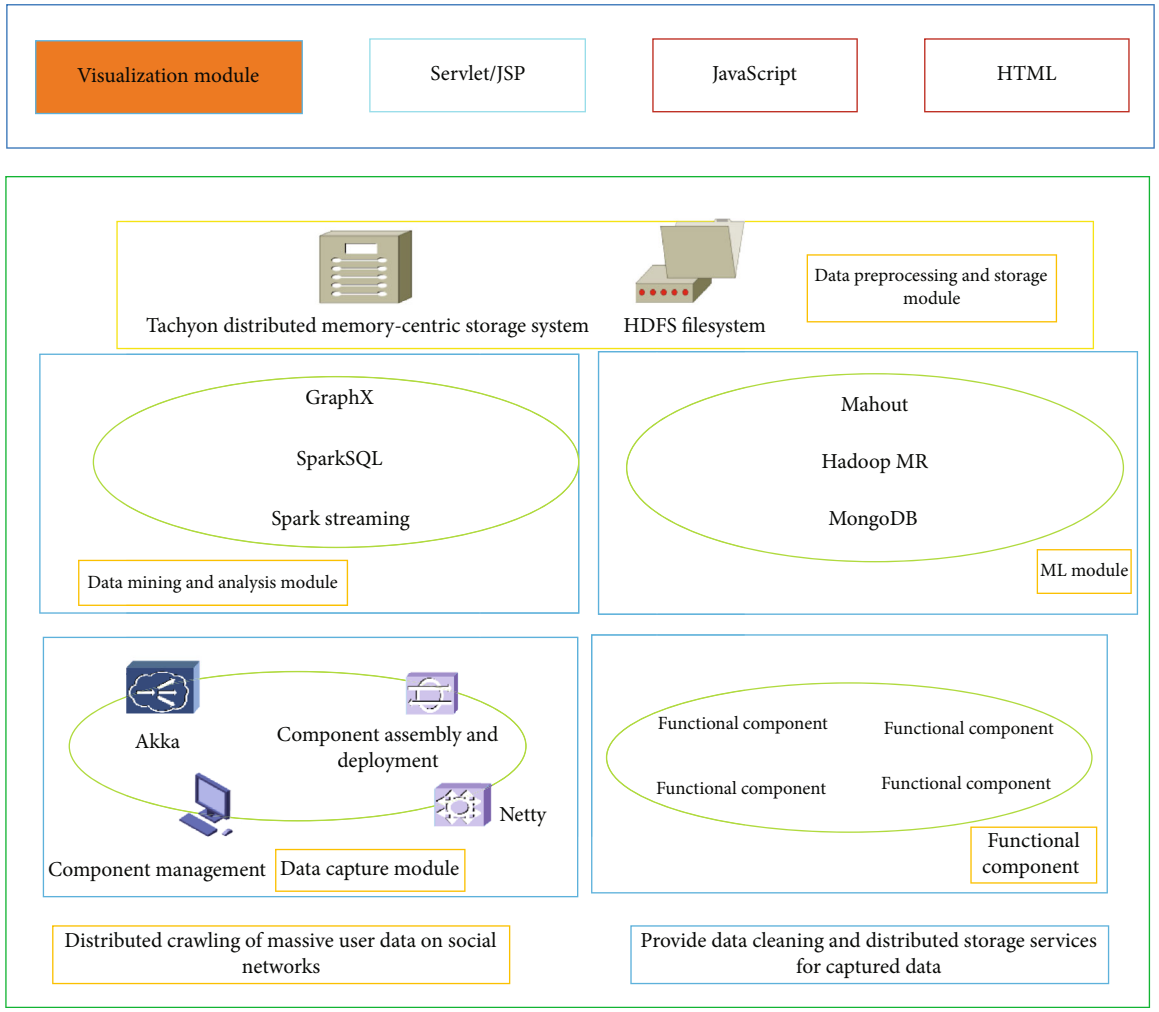


Figure 1: Social network big data analysis platform architecture.

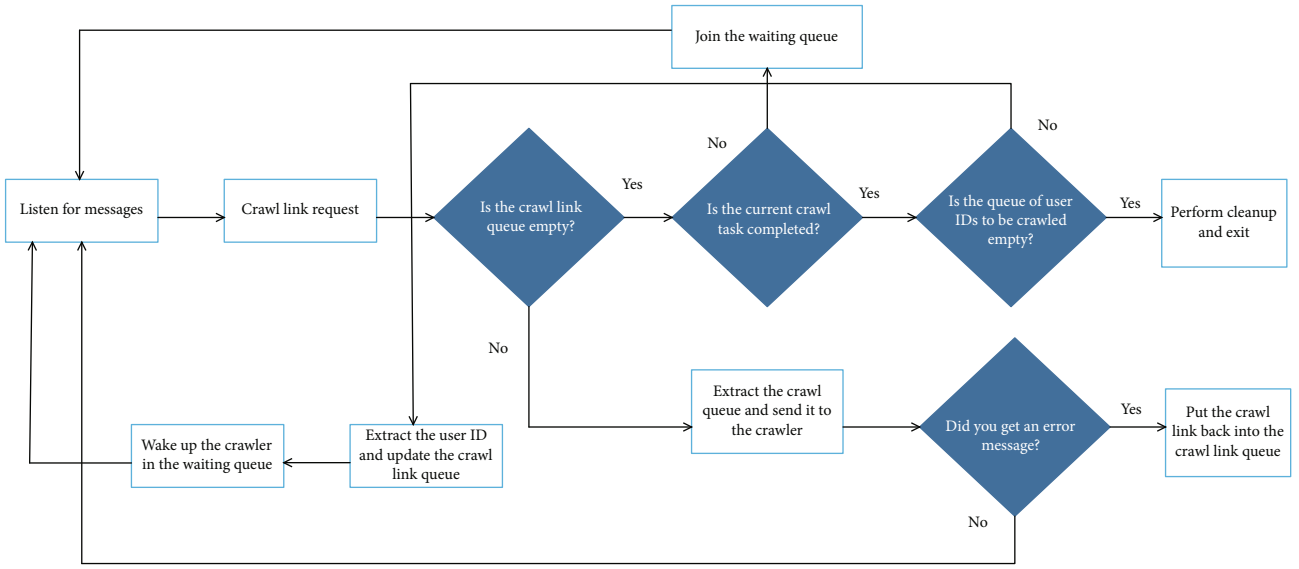


Figure 2: The crawler crawling process.

It can run on a cluster of a large number of cheap disks and provide reliable data slice storage and management, while ensuring faster data read and write speeds. The data preprocessing and storage module is basically the same as HDFS in function, so this module is mainly realized by building HDFS.

In order to ensure the versatility of the data analysis platform, so that the large amount of microblog data captured can serve more data mining and machine learning algorithms, the data preprocessing and storage module does not specify special features when storing the large amount of microblog data. Before the data is merged, the data needs to be formatted. The main work is to regularize the time field in Weibo and complete the default field to make it meet the structure of the database table. This ensures that upper-level applications can directly query text data through SQL (Spark SQL or Hive on Spark) when manipulating data.

The core component of the social network big data analysis platform is a data mining and analysis module built with Spark as the core. The purpose of this module is to use Spark's fast distributed computing capabilities and the MLlib machine learning components and GraphX graph computing components already provided by Spark. The machine learning and data mining algorithms implemented by the users themselves process the massive microblog data in the platform data storage module to complete the social network data mining and analysis tasks in the big data environment. All data mining and analysis tasks in this module will eventually be submitted to Spark to run in the form of Spark jobs. At present, there are two ways to run Spark: one is interactive operation in spark-shell, and the other is submitted through the Jar package mode to run offline. At the same time, Spark can also be used as a data source to read data from the outside using SQL statements through the JDBC interface.

When the social network big data analysis platform in this paper performs data processing and analysis tasks, Spark-JobServer will submit the Jar package to the deployed Spark cluster for execution. At the same time, in order to make the front-end more convenient to display the data, when the data visualization module requests the data in the platform, the data mining and analysis module will be used as the data source interface to complete the data query service through Hive on Spark.

From the user's point of view, there is no difference whether Hive is based on Spark or Hadoop, and data operations can be performed by connecting to the JDBC interface through the same statement. In this paper, the social network big data platform needs to query and access the data of the data preprocessing and storage module when data visualization. These operations will be implemented through the JDBC interface of Hive on Spark in the SQL interaction mode of the traditional relational database. The workflow of using Hive on Spark is shown in Figure 3.

4. Improved Differential Privacy Algorithm

4.1. Principle Analysis of Differential Privacy. Differential privacy initially achieved good results in the application of database statistical information. The differential privacy method is developed based on the concept of "neighbors" data sets. The so-called "neighbor" data set concept refers to a data set that is different from the original data set, and the difference is only one record. This concept gives the most stringent definition of the differential privacy method, ensuring that the differential privacy method can resist attackers to the greatest extent and prevent the leakage of private data. The advantages of differential privacy make it highly regarded.

4.1.1. Problem Description. In order to describe the privacy protection problem more vividly, this article uses a connected undirected graph G to represent a social network, where each node of the graph represents an object in the social network, and each undirected edge of the graph represents the relationship between two objects. The mathematical notation is as follows:

In the graph $G(V, E)$, V is the set of all nodes, and E is the set of all edges; the node set V is

$$V = \{v_i | i = 0, 1, 2, 3, \cdots, n-1\}, \tag{1}$$

where n is the number of nodes. The edge set is

$$E = \left[\left(v_i, v_j\right) | i, j = 0, 1, 2, 3, \cdots, m-1, i \neq j\right]. \tag{2}$$

Let q denote a query function of the graph and $q(G)$ denote the result of the query function q acting on the graph G.

As we all know, when users use the query function q to obtain information from the social network graph G, the real result should be $q(G)$. In order to protect the private data on social networks, the real results cannot be returned to the user. This article needs to add some noise with a specific distribution on the basis of $q(G)$, so that the real information will not be leaked, and there is not much discrepancy between the real statistical results and does not affect the normal needs of users. This is the most basic principle of this article to achieve privacy protection.

4.1.2. Definition and Implementation of Differential Privacy. Traditional privacy protection methods often rely on the attacker's knowledge background, which causes them to have different flaws, which are only suitable for specific environments, and cannot achieve satisfactory results when acting on complex social networks. The proposed differential privacy protection method model perfectly solves the lack of privacy protection in social networks. It defines an extremely strict attack model, which realizes the protection of privacy by adding noise to the original data, the conversion of the original data, or the statistical results. Even if the attacker already knows all other data except the target data, the differential privacy mechanism can still achieve a good protection effect, ensuring that the target data will not be leaked.

D_1 and D_2 are two data sets, and only one data is different between them. q represents a random function, $\Pr[A]$ represents the probability of data being leaked, and S is a subset of all values of the function q, if the random function q satisfies

$$\Pr[q(D_1) \in S] < e^{\varepsilon} \cdot \Pr[q(D_2) \in S]. \tag{3}$$

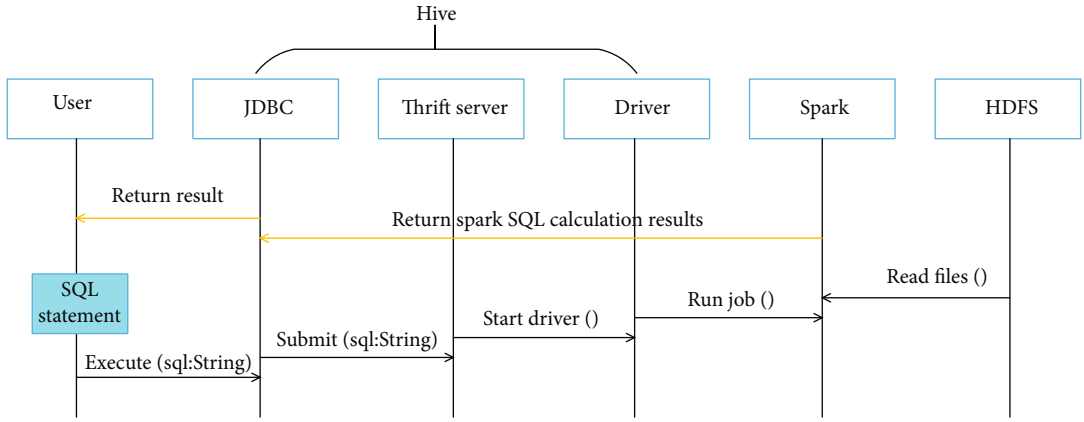


Figure 3: Hive on Spark workflow.

Then, it is said that q provides ε differential privacy protection. There are differences between data sets D_1 and D_2, and there is only one piece of data. This article refers to the original set and neighbor set, respectively. It can be seen that the realization of differential privacy is closely related to D_1 and D_2. After research, it is found that differential privacy can be achieved by adding Laplacian distributed noise.

We call the distribution of the probability density function the Laplace distribution. The parameter μ represents the position, and the parameter $b > 0$ represents the scale.

$$f(x|u,b) = \frac{1}{2} \cdot b \cdot \begin{cases} \exp\left(\frac{u-x}{b}\right) & x \geq u, \\ \exp\left(\frac{x-u}{b}\right) & x < u. \end{cases} \tag{4}$$

In the Laplace distribution, this paper defaults $\mu = 0$, so the only uncertain parameter is b. And b is set based on the difference between the original set and its neighbor set.

For the query function q, this paper calls the difference between the original set and its neighbor set the sensitivity. Its sensitivity is defined as

$$\Delta q = \max\ [D_1 \quad D_2 \quad q(D_1) - q(D_2)]. \tag{5}$$

So far, the basic concepts related to differential privacy have been introduced. Its working mechanism can be understood as follows: a data set D, a query function q, and privacy control parameters ε are input. Based on Laplace's noise, the probability distribution is

$$p(x|\lambda) = \frac{1}{2} \cdot \frac{e^{-x/\lambda}}{\lambda}. \tag{6}$$

Among them, λ is jointly determined by the sensitivity Δq and the control parameter ε, and the sensitivity Δq is related to the data set D and its neighbor set. This working mechanism of differential privacy ensures that the attacker cannot accurately estimate the true "existence" of the target record even if he knows all the records except the target record and prevents the leakage of private data to the greatest extent.

4.1.3. Analysis of the Effectiveness of Differential Privacy. Through the analysis of the working mechanism of differential privacy, this article knows that the Laplacian noise mechanism guarantees the correctness of differential privacy. In the following, the subject thoroughly demonstrates the feasibility of Laplace noise and the effectiveness of the differential privacy protection method.

For the data set D and the query function q acting on D, the mechanism A is

$$A(D) = q(D-1) + \text{Laplace}\left(\frac{\Delta q}{\varepsilon}\right). \tag{7}$$

The Laplacian noise mechanism satisfies the protection conditions of ε differential privacy and ensures the effectiveness of differential privacy. The goal of differential privacy is to maximize query accuracy and minimize the risk of privacy leakage. The noise of the Laplacian distribution makes the user or the attacker get a similar output for any result of the query function, regardless of whether the data set contains or does not contain the target record.

4.2. Improved Differential Privacy. The unique advantages of differential privacy make it attract strong attention as soon as it is proposed. Although the traditional differential privacy protection method can achieve good protection effects in the application of database statistical information, the complexity and data relevance of the social network itself hinder the application of the traditional differential privacy protection method. In response to this, this paper proposes an interactive differential privacy mechanism, so that the differential privacy protection mechanism can be better applied to social networks and better protect people's private data on social networks.

4.2.1. Problem Description. The key issue of differential privacy protection in social networks is to compare the original graph and its "neighbor graph" to calculate the sensitivity and adjust the parameter ε to control privacy leakage. The traditional differential privacy protection model can get good results for individual users' independent queries (that is, there is no relationship between each query of the user),

but when the queries are closely related, it is difficult to avoid the leakage of private data. There are two main problems encountered in the application of differential privacy on social networks: First, on a graph such as a social network, the points or edges are all related to each other. When changing a point or an edge, the impact of changing a point or an edge is not only the independent point or this edge but also the point near this point or the edge near this edge and even the entire network. Second, under interactive conditions, there may be a relationship dependency between each query of the user. The current query result will depend on all previous query results. At this time, the sensitivity calculation needs to consider the current relationship between the results of each query while considering the impact of the query.

Summarizing the problems of traditional differential privacy applications in social networks, this article found a common phenomenon, and the most important point is that when comparing the original graph and its "neighbor graph" on social networks to calculate the sensitivity, it is not only necessary to calculate the current query situation, but it also needs to calculate the situation related to the current query. In this paper, this relationship factor between each other is called the interaction factor, and it is represented by t.

4.2.2. Definition and Implementation of Interactive Differential Privacy. Interactive differential privacy is implemented on the basis of traditional differential privacy. The main application object is social networks. The goal is to use social networks in a complex environment where "data" is closely related. Even if the attacker already knows all the other data except the target data, it can still achieve a good protection effect and ensure that the target data will not be leaked.

This paper still uses connected undirected graph G to represent a social network, and Q represents a query function for "social network." Assuming that the user is currently querying the social network for the $t+1$th time, the real result should be $Q_t + 1(G)$. Under the action of the interaction factor t, the result of the previous t times or $t+1$ times the result of the query will be affected by t. The results are $Q_1(G)$, $Q_2(G), \cdots, Q_t(G)$; then, $Q_t + 1(G)$ will depend on $Q_1(G)$, $Q_2(G), \cdots, Q_t(G)$. The interactive differential privacy protection mechanism not only needs to calculate the influence of $Q_t + 1(G)$ but also needs to take into account all the influences including $Q_1(G), Q_2(G), \cdots, Q_t(G)$.

G and G' are two social network graphs, Q represents a random function, $\Pr[A]$ represents the probability of data A being leaked, and S is a subset of all values of function Q. Then, as long as the following formula is satisfied, the query function Q realizes the differential privacy protection of ε.

$$\Pr\{[Q_{t+1}(G)|Q_1(G), \cdots, Q_t(G)] \in S\} \leq \exp(\varepsilon) \cdot \Pr\left\{\left[Q_{t+1}\left(G'\right)\middle|Q_1\left(G'\right), \cdots, Q_t\left(G'\right)\right] \in S\right\}. \tag{8}$$

Interactive differential privacy still uses Laplacian distributed noise. Through the new sensitivity calculation formula and constant adjustment of the parameter ε, this article can obtain the Laplacian distribution of the noise that needs to be added and then achieve an interactive differential privacy method. Interactive differential privacy applications on social networks can also maximize query accuracy and minimize the risk of privacy leakage. In fact, although this article defines the dependency relationship between each query of the user, the query function is different for different practical scenarios, which causes the relationship between the query results to be different. Therefore, the interactive differential privacy protection model needs to refine the relationship between query results based on practical indicators and continuously adjust to achieve better protection effects.

4.2.3. Analysis of the Effectiveness of Interactive Differential Privacy. By adding Laplace distributed noise, this paper can still ensure the correctness of interactive differential privacy. For the data set D and the query function Q acting on D, the mechanism A is

$$A(D) = \left[Q(D)_t Q(D)_{t-1} \cdots Q(D)_1\right] - \text{Laplace}\left(\frac{\Delta Q}{\varepsilon}\right). \tag{9}$$

This guarantees $t * \varepsilon$ differential privacy. When the interaction factor is t, and each query Q satisfies the ε/t differential privacy, the mechanism will provide the guarantee of ε differential privacy.

$$A(D) = \left[Q(D-1)_t Q(D-1)_{t-1} \cdots Q(D-1)_1\right] - \text{Laplace}\left(\frac{t \cdot \Delta Q}{\varepsilon}\right). \tag{10}$$

Therefore, the interactive differential privacy mechanism proposed in this paper can also achieve ε differential privacy by adding Laplacian distributed noise, maximize query accuracy, and minimize privacy leakage, while solving the complexity and data relevance of social networks.

4.2.4. Algorithm Flow of Interactive Differential Privacy Mechanism. The interactive differential privacy mechanism is implemented by adding Laplacian distributed noise. Unlike traditional differential privacy, in the sensitivity calculation process of interactive differential privacy, in addition to considering the impact of the current query, this article also needs to consider the interaction factor. The algorithm of the interactive differential privacy mechanism mainly includes two parts: the sensitivity solving process and the noise adding process. The flow of the interactive differential privacy algorithm is shown in Figure 4.

5. Experimental Analysis

5.1. Experimental Design and Implementation. This experiment will simulate the evolution process of the proportion of users who take privacy protection behaviors in the network on four sets of data sets and get the final evolution result by setting different costs and benefits. We compare the evolution result with the theoretical derivation to prove the correctness of the model. Three sets of data are real social

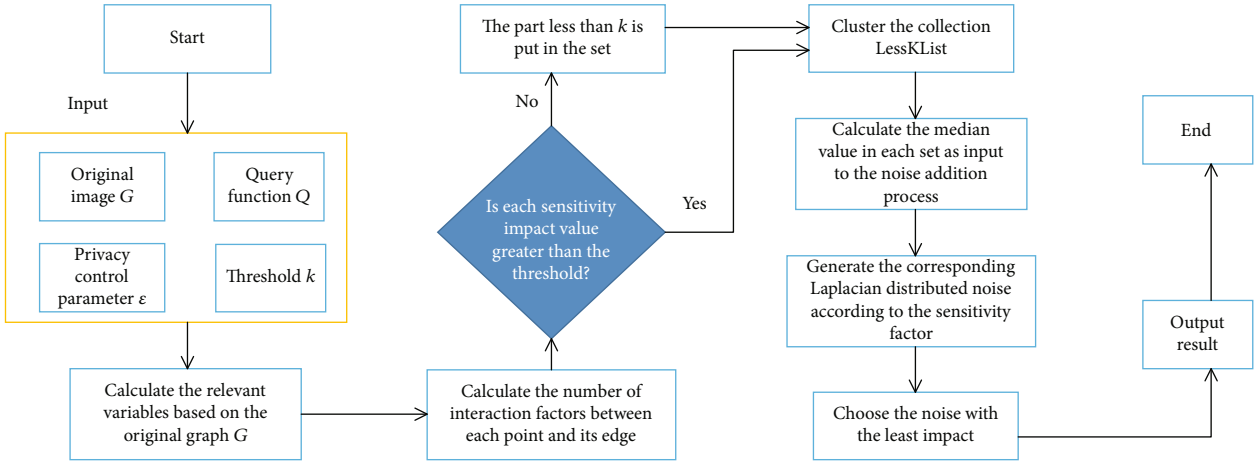


Figure 4: Interactive differential privacy algorithm flow.

network structure data, and the source is the Stanford large-scale network data collection SNAP project. The remaining set of data is man-made data, and the structure is a regular graph, that is, each point has the same degree. The data characteristics are shown in Figure 5.

5.2. Analysis of Simulation Results. In real social networks, most of the personal information is still in the hands of users, and connected friends only have a small part of the associated data, so the value of the correlation factor α will not be very large, so we set α to 0.1. Since evolutionary games occur under weak selection conditions, we set w to 0.001.

The evolution process of Ego-Facebook data set users' privacy protection behaviors under different benefit-cost ratios is shown in Figure 6. The proportion of users who initially adopt the privacy protection mechanism is set to $p0 = 0.4$. Since the evolution process is relatively slow, in order to clearly show the data, the data points are average values. Although the abscissa of each data set is from 0 to 10000, each data point is the average of the previous N results. The number of updates is $10000 \times N$. Where $c = 1$, b changes to achieve different benefit-cost ratios. According to the network structure and parameter settings, the theoretical result is that when $b/c > 0.99$, the proportion of users who adopt the privacy protection mechanism is finally 1. When $b/c < 0.99$, no one will finally adopt the privacy under the benefit-cost ratio protection mechanism. The simulation results show that the model can effectively evolve the dynamic evolution of user privacy protection behavior in social networks and can help social network managers design effective security services based on factors such as network structure, benefits, and costs. It can not only improve the security of the network but also reasonably calculate its own development costs.

When a social platform launches a new privacy protection setting or a user uses a certain privacy protection setting of the platform for the first time, there is no cost-benefit ratio that can be referred to in the past for strategy learning. At this time, users will decide whether to adopt it according to their privacy habits. Therefore, in the early stages of evolution, social networks had a proportion of initial users who adopted

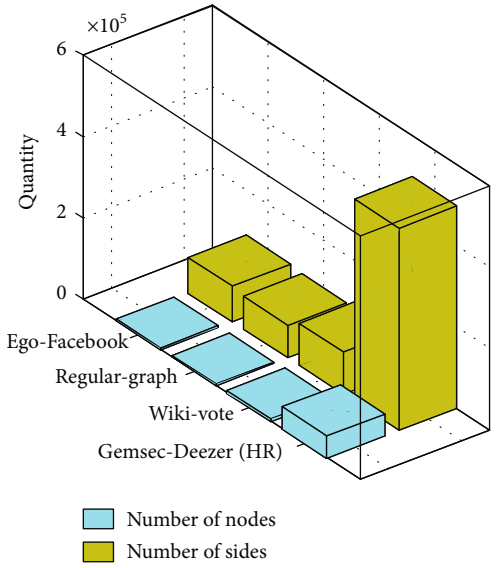


Figure 5: Simulation data set of user privacy behavior evolution process.

privacy protection settings. In order to verify the correctness, Figure 7 shows the simulation results of setting four different initial values $p0 = \{0.3, 0.4, 0.5, 0.6\}$ on the Ego-Facebook data set. Among them, the cost-benefit ratio in Figure 7(a) is $c = 1.2$, and the cost-benefit ratio in Figure 7(b) is $c = 0.8$.

5.3. Comparison of Simulation Results of Each Data Set. In order to illustrate the correctness of the model, this experiment conducted simulation experiments on four data sets with different structures. Figures 8–10 are the evolution results of regular-graph, Wiki-vote, and gemsec-Deezer (HR) data sets of user privacy behavior. From the experimental results, it can be seen that the curve of $b/c > 1$ finally converges to 1, and the curve of $b/c < 1$ finally converges to 0. When the average network degree approaches infinity, the critical value approaches 1. The experimental results prove that the model is feasible. At the same time, it is found that as the number of nodes in the network increases, the evolution process becomes slower and slower. The data set used

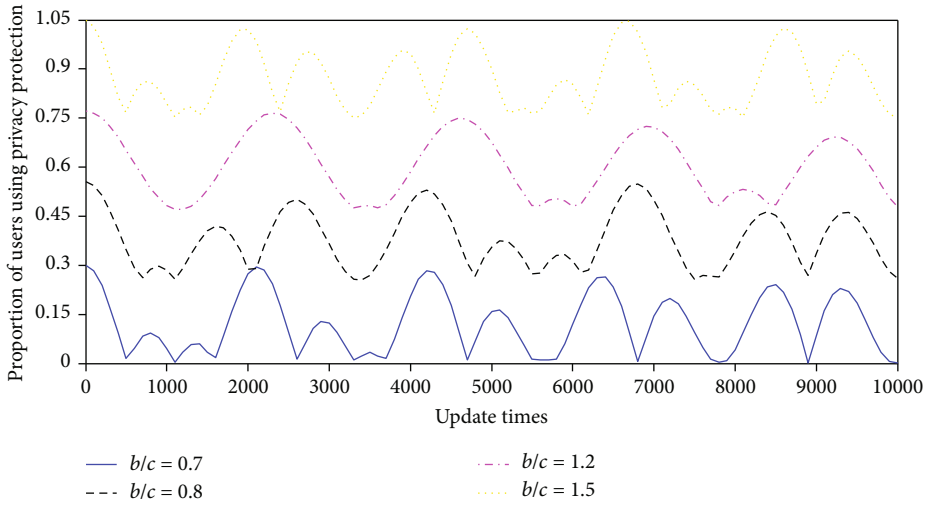
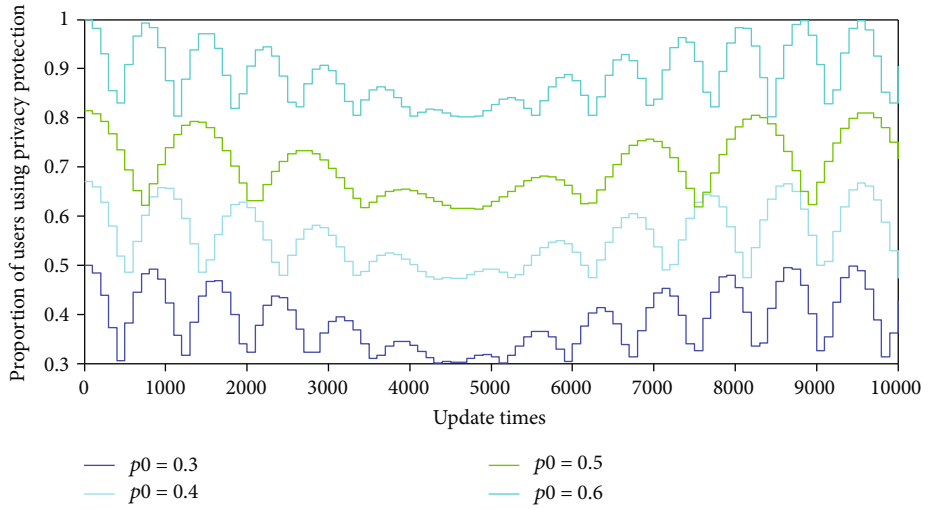
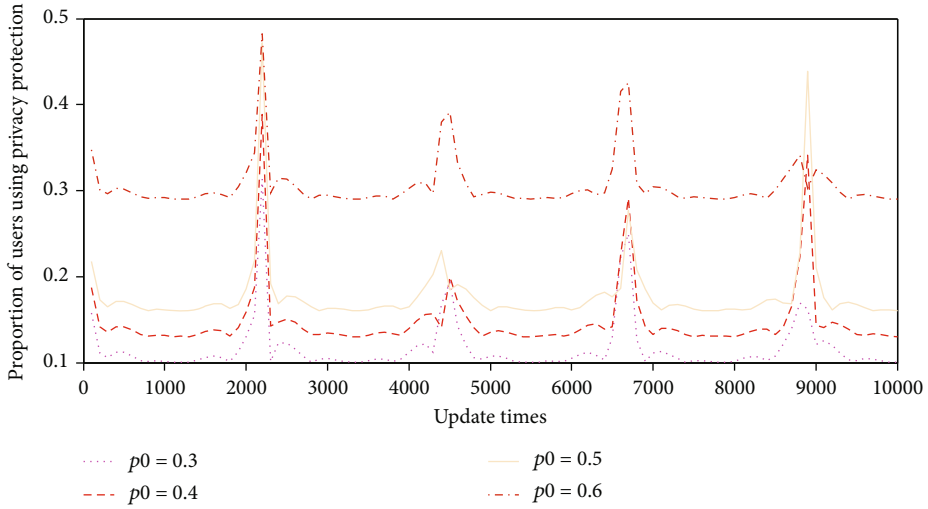


Figure 6: Ego-Facebook data set privacy protection behavior evolution simulation results.



(a) The cost-benefit ratio is $c = 1.2$



(b) The cost-benefit ratio is $c = 0.8$

Figure 7: Evolution results of different initial user proportions.

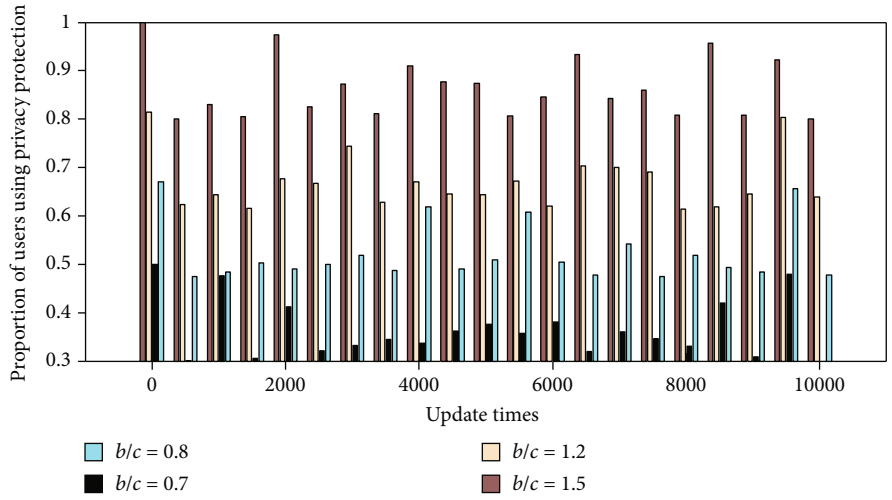


Figure 8: The evolution result of the privacy protection behavior of the regular-graph data set.

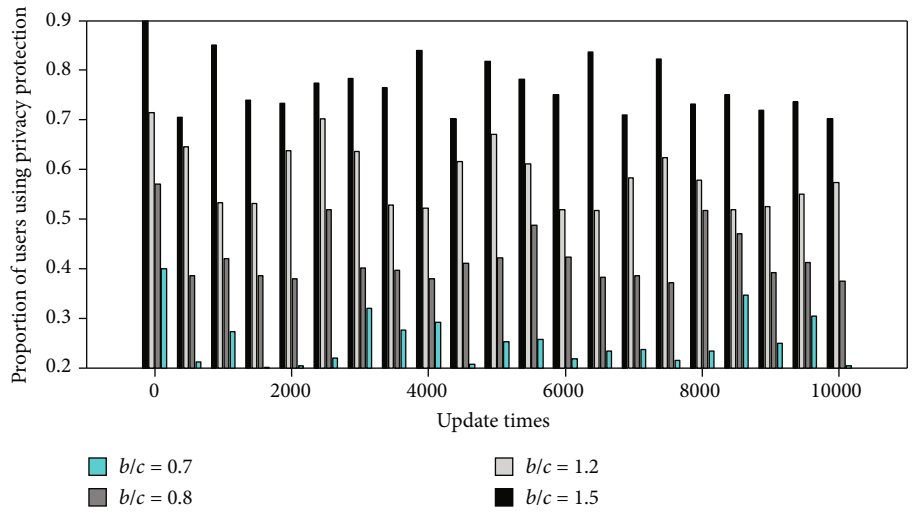


Figure 9: Wiki-vote data set privacy protection behavior evolution result.

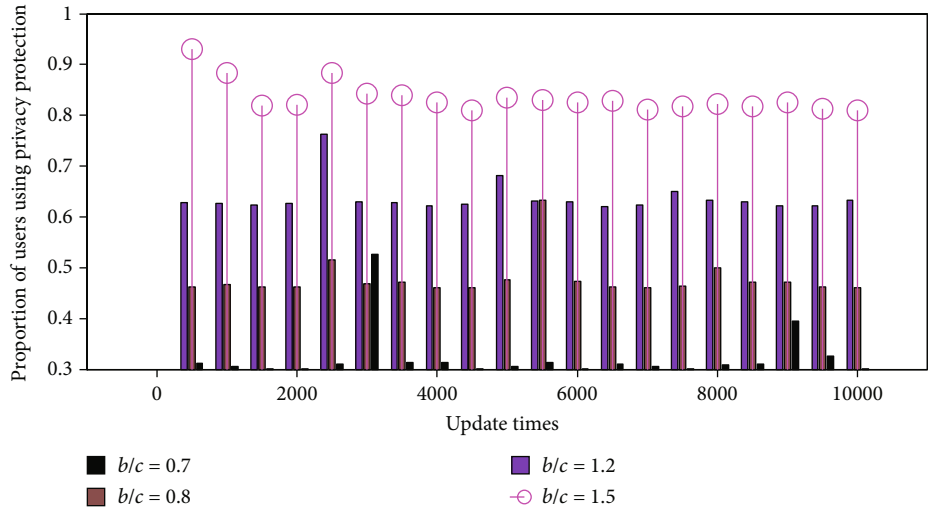


Figure 10: The evolution result of the privacy protection behavior of the gemsec-Deezer (HR) data set.

in Figure 8 is an artificial data set, and the number of nodes is the same as the Ego-Facebook data set, but the connection structure is different. The update time required for the regular structure to converge to the result is longer than the time required for the actual heterogeneous social network structure, indicating that the heterogeneity of the network structure can accelerate the curve convergence. It can be found that the evolution process varies with the network structure, but the evolution trend follows the theory derived from the model, which verifies the correctness of the model. Experimental results show that the model can effectively model the dynamic evolution process of user privacy protection behavior in social networks.

6. Conclusion

Aiming at the status quo that traditional differential privacy methods cannot be applied to interactive social networks, the subject research first designed a novel differential privacy protection mechanism based on interaction factors. Relying on this new mechanism, this paper can not only prevent the leakage of private data to the greatest extent as the traditional differential privacy method but also solve the problem of privacy leakage caused by the characteristics of the data relevance of the social network itself. This paper proposes a user security behavior game model based on network evolutionary game theory and defines the basic elements of the game model and the updated rules of the evolutionary game. The dynamic equation is derived based on the game model, which shows the evolution process of the benefit-cost ratio of the user's security behavior under the social network structure with the privacy protection provided by the platform. In order to verify the effectiveness of the model, simulation experiments were designed on artificial data sets and real data sets. Experimental results show that the model can effectively describe the evolution of user safety behavior. In the social attribute privacy protection scenario, although the method proposed in this paper solves the privacy problem of social attribute leakage that may occur during access control in social networks, there is also social attribute privacy leakage in other scenarios, such as users in data mining. For sensitive attribute leakage, how to protect user attribute privacy while ensuring data availability is the next problem to be solved.

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest

We declare that there is no conflict of interest.

References

[1] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 591–606, 2018.

[2] R. Chen, B. C. Fung, P. S. Yu, and B. C. Desai, "Correlated network data publication via differential privacy," *VLDB Journal*, vol. 23, no. 4, pp. 653–676, 2014.

[3] J. M. Such and N. Criado, "Resolving multi-party privacy conflicts in social media," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1851–1863, 2016.

[4] Q. Fang, J. Sang, C. Xu, and M. S. Hossain, "Relational user attribute inference in social media," *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 1031–1044, 2015.

[5] P. Wang, Z. Gao, X. Xu, Y. Zhou, H. Zhu, and K. Q. Zhu, "Automatic inference of movements from contact histories," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 386-387, 2010.

[6] E. K. Wang, Y. Li, Y. Ye, S. M. Yiu, and L. C. K. Hui, "A dynamic trust framework for opportunistic mobile social networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 319–329, 2018.

[7] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[8] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "My privacy my decision: control of photo sharing on online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 199–210, 2017.

[9] R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Preventing private information inference attacks on social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1849–1862, 2013.

[10] M. Vegni and V. Loscri, "A survey on vehicular social networks," *IEEE Communication Surveys and Tutorials*, vol. 17, no. 4, pp. 2397–2419, 2015.

[11] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, "Social network security: issues challenges threats and solutions," *Information Sciences*, vol. 421, pp. 43–69, 2017.

[12] E. Palomar, Á. Galán, A. Alcaide, and L. González-Manzano, "Implementing a privacy-enhanced attribute-based credential system for online social networks with co-ownership management," *IET Information Security*, vol. 10, no. 2, pp. 60–68, 2016.

[13] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: optimal estimation and privacy analysis," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5677–5690, 2018.

[14] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377–388, 2014.

[15] B. Zhou and J. Pei, "The k -anonymity and l -diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 47–77, 2011.

[16] H. Hu, G.-J. Ahn, and J. Jorgensen, "Multiparty access control for online social networks: model and mechanisms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1614–1627, 2013.

[17] B. Carminati, E. Ferrari, and M. Viviani, "Security and trust in online social networks," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 4, no. 3, pp. 1–120, 2013.

[18] Z. Zhang, Q. Gu, T. Yue, and S. Su, "Identifying the same person across two similar social networks in a unified way: globally and locally," *Information Sciences*, vol. 394, pp. 53–67, 2017.

[19] J. M. Such and M. Rovatsos, "Privacy policy negotiation in social media," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 11, no. 1, pp. 1–29, 2016.

[20] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy leakage of location sharing in mobile social networks: attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 646–660, 2018.